Sound generation based on image color spectrum with using the recurrent neural network

N.A.Nikitin[1], V.L.Rozaliev[1],Yu.A. Orlova[1], A.V. Alekseev[1]

[1]Volgograd State Technical University, Lenin avenue 28,Volgograd, Russia, 400005

Abstract. This work is devoted to development and approbation of the program for automated sound generation based on image colour spectrum with using the recurrent neural network. The work contains a description of the transition between colour and music characteristics, the rationale for choosing and the description of a recurrent neural network. The choices of the neural network implementation technology as well as the results of the experiment are described.

Keywords: artificial neural networks, recurrent neural network, long-short term memory, Newton correlation scheme.

1. Introduction

Since the music began to be recorded on paper in the form of musical notation, the original "ways" of its composition began to appear. One of the first methods of algorithmic composition was the method of composing music invented by Mozart - "The Musical Game of the Dice". The first computer musical composition - "Illiac Suite for String Quartet" - was created in 1956 by the pioneers of using computers in music - Lejaren Hiller and Leonard Isaacson [1]. In this work, almost all the main methods of algorithmic musical composition are used: probability theory, Markov chains and generative grammars.

The development of computer music, including the sound generation by image, in the last century was severely limited by computing resources - only large universities and laboratories could afford to buy and hold powerful computers, and the first personal computers lacked computing power. However, in the 21st century, almost everyone can study computer music.

Now, computer music can be used in many industries: creating music for computer games, advertising and films. Now, to create background music compositions in computer games and advertising, companies hire professional composers or buy rights to already written musical works. However, in this genre the requirements for musical composition are low, which means that this process can be automated, which will allow companies to reduce the cost of composing songs. Also, the generation of sounds based on image can be applied in the educational process [2]. The development of musical perception in preschool children can be in the form of integrated educational activities, which is based on combinations of similar elements in music and arts (the similarity of their mood, style and genre) [3].

The greatest success of the theory of automation of the process of writing and creating music made up relatively recently (at the end of XX century), but mostly associated with the study and repetition of different musical styles [4].

Since the process of creating music is difficult to formalize, artificial neural networks are best suited for automated sound generation – they allow identifying connections that people do not see [5]. In addition, to reduce the user role in the generation of music, it was decided to take some of the musical characteristics from the image. Thus, the purpose of this work is to increase the harmony and melodicity of sound generation based on image colour spectrum through the use of neural networks.

To achieve this purpose the following tasks were identified:

- Determine the correlation scheme between colour and musical characteristics.
- Review the types of neural networks and choose the most suitable type for generating musical compositions.
- Describe the neural network used to generate music compositions by image.
- Choose neural network implementation technology.
- Choose a method for sounds synthesizing.
- Design and develop a program for sound generation using neural networks.
- Make an experiment to assess the harmony and melody of the output musical composition.

2. From colour to musical characteristics

To reduce the user role in the generation of music, some of the musical characteristics are obtained by analysing the colour scale of the image. Thus, the character of the output musical composition will correspond to the input image. This feature makes possible to use this approach for creating background music in computer games, advertising and films.

The key characteristics of a musical work are its tonality and tempo. These parameters are determined by analysing the colour scheme of the image. To begin with, we determine the ratio of colour and musical characteristics [6] (table 1).

Table 1. Correlation between color and musical characteristics.

Colour characteristics	Musical characteristics
Hue (red, blue, yellow...)	Pitch (c, c#, d, d#, …)
Colour group (warm/cold)	Musical mode (major/minor)
Brightness	Octave
Saturation	Duration

Then, it is necessary to define the correlation scheme between the color and pitch name. At the moment, there are a large number of such schemes, but in this work, was chosen the Newton scheme (table 2).

Table 2. Correlation scheme between color and pitch names.

Colourname	Pitch
Red	C
Red – orange	C#
Orange	D
Yellow – orange	D#
Yellow	E
Green	F
Green - blue	F#
Blue	G
Blue-violet	G#
Violet	A
Yellow-green	A#
Pink	H

As can be seen from Table 1, the tonality of a composition is determined by two colour characteristics - a hue and a colour group, the tempo by brightness and saturation. The algorithm for determining the tonality relies on image analysis and table 1. It consists of 3 steps and described below.

Step 1. Converting the input image from RGB to HSV colour space. This step allows transforming the image to a more convenient form, because HSV space already contains the necessary characteristics - the name of the colour (determined by the parameter hue), saturation and brightness (value parameter).

Step 2. Analysing the whole image and determining the predominant colour.

Step 3. Determining the name and colour group of predominant colour.

Step 4. According to table 1 and table 2 define the tone of the musical composition (pitch and the musical mood).

To determine the tempo of composition, it`s necessary to get the brightness and saturation of predominant color, and calculate the tempo, according to these parameters.

3. The choice of a neural network to generate musical compositions

An important feature of feedforward neural networks is that, this neural network has a common limitation: both input and output data have a fixed, pre-designated size, for example, a picture of 100 × 100 pixels or a sequence of 256 bits. A neural network, from a mathematical point of view, behaves like an ordinary function, albeit very complexly arranged: it has a pre-defined number of arguments, as well as a designated format in which it gives the answer.

The above features are not very difficult if we are talking about the pictures or pre-defined sequences of symbols. But for the processing of any conditionally infinite sequence in which it is important not only the content but also the order of information, for example, text or music, neural networks with feedback should be used - recurrent neural networks (RNN). In recurrent neural networks neurons exchange information among themselves: for example, in addition to a new piece of incoming data the neuron also receives some information about the previous state of the network. Thus, the network realizes a "memory" which fundamentally changes the nature of its work and allows to analyse any data sequences in which it is important the order of information [7].

However, the great complexity of RNN networks is the problem of explosive gradient, which consists in the rapid loss of information over time. Of course, this only affects the weights, not the states of the neurons, but it is in them information accumulates. Networks with long-short term memory (LSTM) try to solve this problem through the using filters and an explicitly specified memory cell. Each neuron has a memory cell and three filters: input, output and forgetting. The purpose of these filters is to protect information. The input filter determines how much information from the previous layer will be stored in the cell. The output filter determines how much information the following layers will receive. Such networks are able to learn how to create complex structures, for example, compose texts in the style of a certain author or compose simple music, but they consume a large amount of resources [8].

Thus, to implement the program for automated sound generation based on image colour spectrum, it is necessary to use recurrent neural networks with long short-term memory - RNN LSTM. This kind of neural networks is used to generate musical compositions in various programs such as Magenta. Magenta is an open source music project from Google. Also, RNN LSTM is used in BachBot. This is the program that creates the musical composition in the Bach style. And this kind of neural network is used in DeepJaz - the system that allows to generate jazz compositions based on analysis of midi files [9].

4. Description of the used neural network

Recurrent neural network (RNN) has recurrent connections which allow the network to store information related to the inputs. These relationships can be considered similar to memory. RNN is especially useful for the study of sequential data, such as music.

In Tensor Flow, the repeated connections on the graph are deployed into an equivalent feed forward neural network. Then this network is trained using the technique of gradient descent, called back propagation through time (BPT).

There are a number of ways in which RNN can connect to itself with cyclic compounds. The most common are networks with long-short term memory (LSTM) and gated recurrent units (GRU). In both cases, networks have multiplicative neurons that protect their internal memory from overwriting, allowing neural networks to process longer sequences. In this work, LSTM is used. All recurrent neural networks have the form of a chain of repeating modules. In standard RNNs, this repeating module will have a very simple structure, for example, one layer of tanh. LSTMs also have this chain, but the repeating module has a more complex structure. Instead of having one layer of the neural network, there are four interacting with each other in a special way [10].

The first step in LSTM is to decide what information we are going to throw out of the cell state. This decision is taken by the sigmoid layer. This layer looks at the value of h_t-1 output and x_t input, calculates a value in the range from 0 to 1 for each C_{t-1} state. If the layer returned 1, this means that this value should be left (remember), if 0 - removed from the state of the cell. For example, in the state of a cell, the characteristics of the current measure can be stored - if the measure is not yet complete, then it is necessary to leave the characteristics in memory, if work is already in progress, then new parameters must be memorized.

The next step is to decide what new information we are going to store in the state of the cell. To do this, firstly the sigmoid layer decides what values we will update. Next, the tanh layer creates a vector of new candidate values, C_t, that can be added to the state.

The next step is to update the old C_{t-1} state in the new C_t state. To do this, it is necessary to multiply the old f_t state, thus deleting the information from the state. Then, it's necessary to add resulting value and i_t * C_t. Thus, we get new candidate values, scaled by the update coefficient value of each state value.

At the last step, we need to decide what will output this layer. This output will be based on the state of the cell. First, we pass the input value through the sigmoid layer, which decides which parts of the cell state should be output. Then, we process the state of the cell using tanh (to shift the value between -1 and 1), and multiply it by the output of the sigmoid layer.

The behaviour of a neural network is determined by the set of weights and displacements that each node has. Therefore, for the properly work of neural network we need to configure them to some correct value. First, it is necessary to determine how good or bad any output is according to the input value. This value is called cost. Once the cost is received, we need to use the back propagation method. In fact, it reduces to calculating the cost gradient relative to the weights (differential of the cost for each weight for each node in each layer), and then it is necessary to use the optimization method to adjust the weights to reduce the cost. In this work, we will use the method of gradient descent.

For the training of a neural network, it is proposed to feed a vector that contains the following parts [11]:

- Note name: MIDI representation of current note. Used to represent the pitch of a note.
- Time when note on.
- Time when note off.
- The velocity of the note playback.

To determine the correct output according to the input, it is suggested to transform the vector as follows: let there be a vector of notes {c, d, e, f, g, a, h}, then the learning vector will be {{c, d}, {d, e}, {e, f}, {f, g}, {g, a}, {a, h}}. This method of learning a neural network is used, for example, to predict time series [12].

5. The choice of technology for the implementation of an artificial neural network

To implement an artificial neural network, the Python programming language was chosen, because the language is cross-platform, it is aimed at improving developer productivity and code readability. In addition, this language is focused on data analysis, and therefore contains a large number of libraries for deep learning.

Theano is an extension of the Python language, which allows to efficiently calculating mathematical expressions containing multidimensional arrays. Since this library is low-level, the process of creating a model and determining its parameters requires writing voluminous and noisy code. However, the advantage of Theano is its flexibility, as well as the availability of the implementation and use of its own components [13].

Tensor Flow is an open source library for numerical calculation using stream graphs. This library, as well as Theano, is a low-level library, which means that the development process is complex. However, due to the low level of development of neural networks, a more flexible model can be obtained. Also the advantage of this library is a large community and good documentation [14].

Lasagne is a lightweight wrapper for the Theano library. Programming with Lasagne is quite low-level - it is necessary to declare each level of the neural network by using modular building blocks over Theano. Lasagne acts as a compromise between the flexibility of Theano and the simplicity of Keras [15].

Keras is a high-level API for the development of neural networks written in Python and capable of running based on Tensor Flow, CNTK or Theano. This library was developed with an emphasis on the possibility of rapid experimentation. The downside of this library is a small flexibility [16].

MXNet is an open source deep learning system used for training and deploying deep neural networks. Since MXNet is a high-level library, the development of neural networks using MXNet is simpler and faster than using Theano or Tensor Flow, but it is inferior to the Keras library due to the large number of supported languages and large scaling possibilities, which makes the program code more cumbersome [17].

To compare the libraries, the following criteria were proposed: flexibility, scalability, parallel computing support, GPU computing support. All considered libraries were estimated according to the above criteria on a five-point scale, where 0 is the minimum value of the criterion, and 5 is the maximum. The results of comparing the libraries are presented in Table 3.

Table 3. Comparison of deep learning libraries.

Parameter	Theano	Tensor Flow	Lasagne	Keras	MXNet
Flexibility	5	4	3.5	2	3
Scalability	4	5	4	5	5
Parallel computing support	5	4	5	5	5
GPU computing support	4	5	4	5	5

Thus, we can conclude, that to develop a recurrent neural network for the generation of musical compositions, we should use the Keras library, since this library allows to work based on Theano and Tensor Flow, taking advantage of them, while the process of developing neural networks using this library is simple and fast, which allows to create prototypes for rapid experimentation.

6. Sound synthesis

In the process of studying the methods of sound synthesis, four most popular methods of synthesis were considered: additive synthesis, FM synthesis, phase modulation and sampling.

Additive synthesis is very difficult to implement, due to the need for separate control of the volume and height of each harmonic, which even a simple timbre consists of dozens [18].

FM - synthesis is well - suited for synthesizing the sound of percussion instruments, the synthesis of other musical instruments sounds too artificial. The main disadvantage of FM synthesis is the inability to fully simulate acoustic instruments [19].

Phase modulation gives a good enough sound, but is very limited, so it's rarely used in practice.

Sampling is used in most modern synthesizers, since it gives the most realistic sound and is fairly simple to implement [20].

Each of the methods has its advantages and disadvantages, but Sampling was chosen as the most suitable method for sound generating based on image colour spectrum. This method gives the most realistic sound of instruments, which is an important characteristic for the program, and this method is relatively easy to implement. The disadvantage of sampling is its limitation, but for the implementation of the program it is not essential, since for the program needs it is not required possibilities of changing the ready-made presets.

7. Description of program for sound generation

To confirm the effectiveness of the proposed algorithms for generating sounds based on image color spectrum, a Python program was developed. At the input the program receives an image that the user loads manually. This method of loading the image was chosen as the most simple, since at this stage it is only a task to prove the effectiveness of the proposed solutions. Subsequently, this program can be further developed, according to the conditions of applicability to specific areas.

After receiving the path to the image from the user, the program loads the image into memory using the capabilities of the OpenCV library. Converting an image to the HSV color space is also done using this library. Then, the image is analyzed, the tonality and tempo of the work are determined.

After determining the tonality and tempo, the program chooses the most appropriate model for the given situation, based on this model the neural network generates (predicts) a musical composition.

8. The experiment

The program for sound generation was trained on 29 compositions by Ludwig van Beethoven. After training, a set of ten test images of different types (abstract images, landscapes, cities and people) was compiled. For all ten images, output musical compositions were prepared and stored (can be found here: https://github.com/NikitaNikitinVSTU/ImageSoundGeneration). These songs have been sent for analysis to 10 experts who were asked to rate each song according to the following criteria:

- Matching character of image (on a five-point scale).
- Realistic sound of an instrument (piano or guitar).
- Melodiousness of the composition.
- The quality of harmony (accompaniment).
- The pleasantness of the melody for the perception.
- Integrity of the composition.
- Realism/artificiality of the composition.

An example of an abstract image is shown in Figure 1, an example of a landscape is shown in Figure 2 and an example of a city is shown in Figure 3

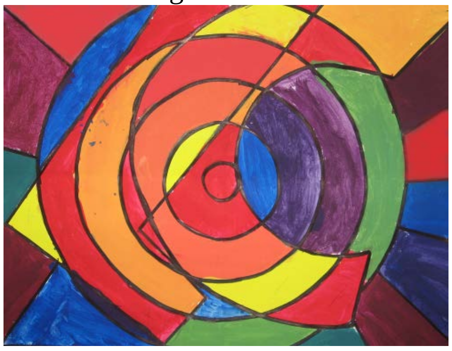

Figure 1. An example of an abstract image.

Figure 2. An example of alandscape.

Figure 3. An example of acity image.

Data from each expert has been processed and analysed (it presented in table 4).

Table 4. Average values of each criterion for all tests of all experts.

Criterion	Average value for all tests
Matching character of image	4.9
Realistic sound of an instrument	3.9
Melodiousness of the composition	4.4
The quality of harmony	4.9
The pleasantness of the melody for the perception	4.6
Integrity of the composition	4.5
Realism/artificiality of the composition	4.3
Matching character of image	4.9
Realistic sound of an instrument	3.9
Melodiousness of the composition	4.4
The quality of harmony	4.9
The pleasantness of the melody for the perception	4.6

Thus, after analyzing the assessments of all experts and calculating the average for each criterion, we can conclude that the piano is heard by experts to sound more realistic than the guitar. It can also be concluded that the composition generated by abstract images is more pleasant by ear than generation by landscape. In general, the overall impression of the generated sounds from experts is positive. Among the minuses, some experts emphasize the uniformity of harmony, sometimes the laceration and lack of realism of the composition, and not enough realism of the guitar.

Making a conclusion on each criterion, it can be said, that all experts rated matching character of image criteria to a high score, according to the second criterion - the piano instrument sounds quite realistic. The melodiousness of the compositions was divided in half, that is, half of the compositions were rated by experts for the top ball, the other half for 4, a generally good result. The quality of harmony was also evaluated by experts at the top ball. The pleasantness of the melody for the perception received 60% of the highest scores and 40% of the quads, which indicates that some compositions sound not quite realistic. The realism and integrity of the compositions is estimated at an average of 4, which is a natural result for computer sound generation.

9. Conclusion

In this work, the scheme of correlation of color and musical characteristics was determined, an overview of the types of neural networks was made and the most suitable type for the generation of musical compositions was selected. Also, the used neural network was described in detail, the technology of implementing the neural network was chosen and the method of synthesis of sounds was chosen. To evaluate the effectiveness of the proposed algorithms, an experiment was conducted to assess the harmony and melodiousness of the output musical compositions.

Analysis of various types and architectures of ANNs concluded that the most suitable network for processing musical information is the recurrent neural networks with long short-term memory (RNN LSTM).

During the description of the used neural network, it was determined that for learning the network it is supposed to input a vector that contains the following parts: MIDI representation of current note, time when note on, time when note off and the velocity of the note playback.

In analyzing the libraries for implementing a neural network in the Python programming language, it was discovered that the Keras library should be used to develop a recurrent neural network, since this library allows to work based on Theano and TensorFlow, taking advantage of them, while the development of neural networks using this library simple and fast, which allows to create prototypes for rapid experimentation.

As a result of the experiment, the model (neural network) on Beethoven's compositions was trained, and compositions of 10 images were generated. These compositions were sent for analysis to experts. As a result of the analysis of expert assessments, it can be concluded that the program generates quite melodic compositions, but it appears that the model was trained on a small number of compositions by only one author.

10. Acknowledgments

The work is partially supported by the Russian Foundation for Basic Research (16-47-340320 and 17-07-01601 projects).

11. References

[1] Ariza, C. Two Pioneering Projects from the Early History of Computer-Aided Algorithmic Composition / C. Ariza // Computer Music Journal. – 2012. – MIT Press. – Vol. 3. – P. 40-56.
[2] Chereshniuk, I. Algorithmic composition and its role in modern musical education / I. Chereshniuk // Art education. – 2015. –Vol 3. – P. 65-68.
[3] Vygotsky, L. Imagination and creativity in childhood / L. Vygotsky //Journal of Russian and East European Psychology. – 2004. – Vol. 42(1). – P. 7-97.
[4] Cope, D. Computer Models of Musical Creativity / D. Cope // MIT Press, Cambridge, Mass. – 2005.
[5] Mazurowski, L. Computer models for algorithmic music composition / L. Mazurowski // Proceedings of the Federated Conference on Computer Science and Information Systems.– Szczecin, Poland, 2012. – P. 733-737.
[6] Caivano, J.L. Colour and sound. Physical and Psychophysical Relations / J.L. Caivano // Colour Research and Application. – 1994. – Vol. 12(2). – P. 126-132.
[7] Sak, H. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition / H. Sak, A. Senior, F. Beaufays //ArXiv e-prints, 2014.
[8] Doornbusch, P. Gerhard Nierhaus: Algorithmic Composition: Paradigms of Automated Music Generation / P. Doornbusch // Computer Music Journal. – 2014. – Vol. 34(3).
[9] Brinkkemper, F. Analyzing Six Deep Learning Tools for Music Generation [Electronic resource]. –URL: http://www.asimovinstitute.org/analyzing-deep-learning-tools-music/.

[10] Mikolov, T. Recurrent neural network based language model / T. Mikolov // Proceedings of Interspeech International Speech Communication Association. – 2010. – Vol. 2010(9). – P. 1045-1048.

[11] Kim, H.K. Transactions on Engineering Technologies: Special Issue of the World Congress on Engineering and Computer Science / H.K. Kim, S.I. Ao, A. Mahyar // Springer Publishing Company, New York, 2013. – P. 796.

[12] Fernandez, J.D. AI Methods in Algorithmic Composition: A Comprehensive Survey / J.D. Fernandez, F. Vico // Journal Artificial Intelligence Research. – 2013. – Vol. 48. – P. 513-582.

[13] Bergstra, J. Theano: a CPU and GPU math expression compiler / J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio // Proceedings of the Python for Scientific Computing Conference (SciPy). – 2010.

[14] Tensor Flow: Large-Scale Machine Learning on Heterogeneous Distributed Systems[Electronic resource]. – URL: https://www.tensorflow.org/.

[15] Lasagne – lightweight library to build and train neural networks in Theano [Electronic resource]. – URL: http://lasagne.readthedocs.org/.

[16] Keras: The Python Deep Learning library [Electronic resource]. – URL: https://keras.io/.

[17] MXNet: A Flexible and Efficient Library for Deep Learning [Electronic resource]. – URL: https:// http://mxnet.io/.

[18] Korvel, G. A Modified Additive Synthesis Method Using Source-Filter Model / G. Korvel, V. Simonyte, V. Slivinskas // Journal of the Audio Engineers Society, Vilnius University Institute of Mathematics and Informatics. – 2015. – Vol. 63(6). – P. 443-450.

[19] Lazzarini, V. Theory and Practice of Modified Frequency Modulation Synthesis / V. Lazzarini, J. Timoney // Sound and Music Technology Group, National University of Ireland. – 2012. – Vol. 58. – P. 459-471.

[20] Russ, M. Sound Synthesis and Sampling / M. Russ // London, Taylor & Francis Group, 2012. – P. 568.